Project name: NFT-based Land Ownership in Minecraft

Team name: 0xLand

Team members: Matt Lefebvre, Munir Baara

GitHub link: https://github.com/mlefeb01/0xLand

Basic Definitions

Minecraft: a 3D (XYZ axis) sandbox game

Block: a 1x1x1 at a specific location in the XYZ plane

Chunk: a 16x16 on the XZ plane (spans the Y plane)

Player: a client that is connected to a Minecraft server instance

ERC721: Non-Fungible Token Standard

Spigot: Minecraft server API

PlayerInteractEvent: Event fired by the SpigotAPI when a player clicks a block

What?

We are going to develop a smart contract (ERC721 compliant) that corresponds to chunk ownership in a Minecraft world. The NFTs will hold XZ coordinates in their metadata.

A Minecraft server plugin will be written using the Spigot server API to query the smart contract and obtain the ownership mapping and will maintain state of the ownership mapping on the server. The plugin will listen for transfer/mint events emitted from the smart contract to maintain state that is consistent with the smart contract. The plugin will have a command for players to "link" (/link <address>) there wallet and associate their account with the provided wallet.

The NFTs will be minted via a react app that allows the user to connect their MetaMask wallet and provide an x/z coordinate of the chunk they want to claim. Chunks will be claimed on a FCFS (first come first serve) basis.

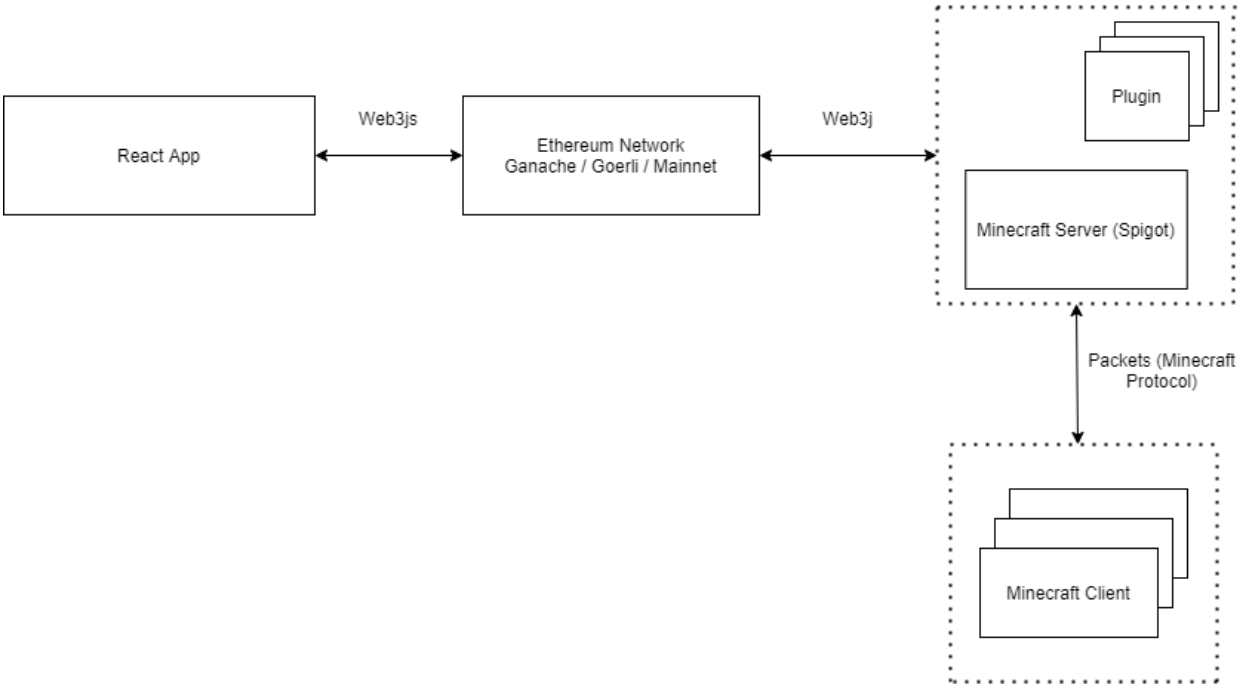
Why?

We are building this because we are interested in gaining more experience with Solidity and believe this is a practical application of NFTs.

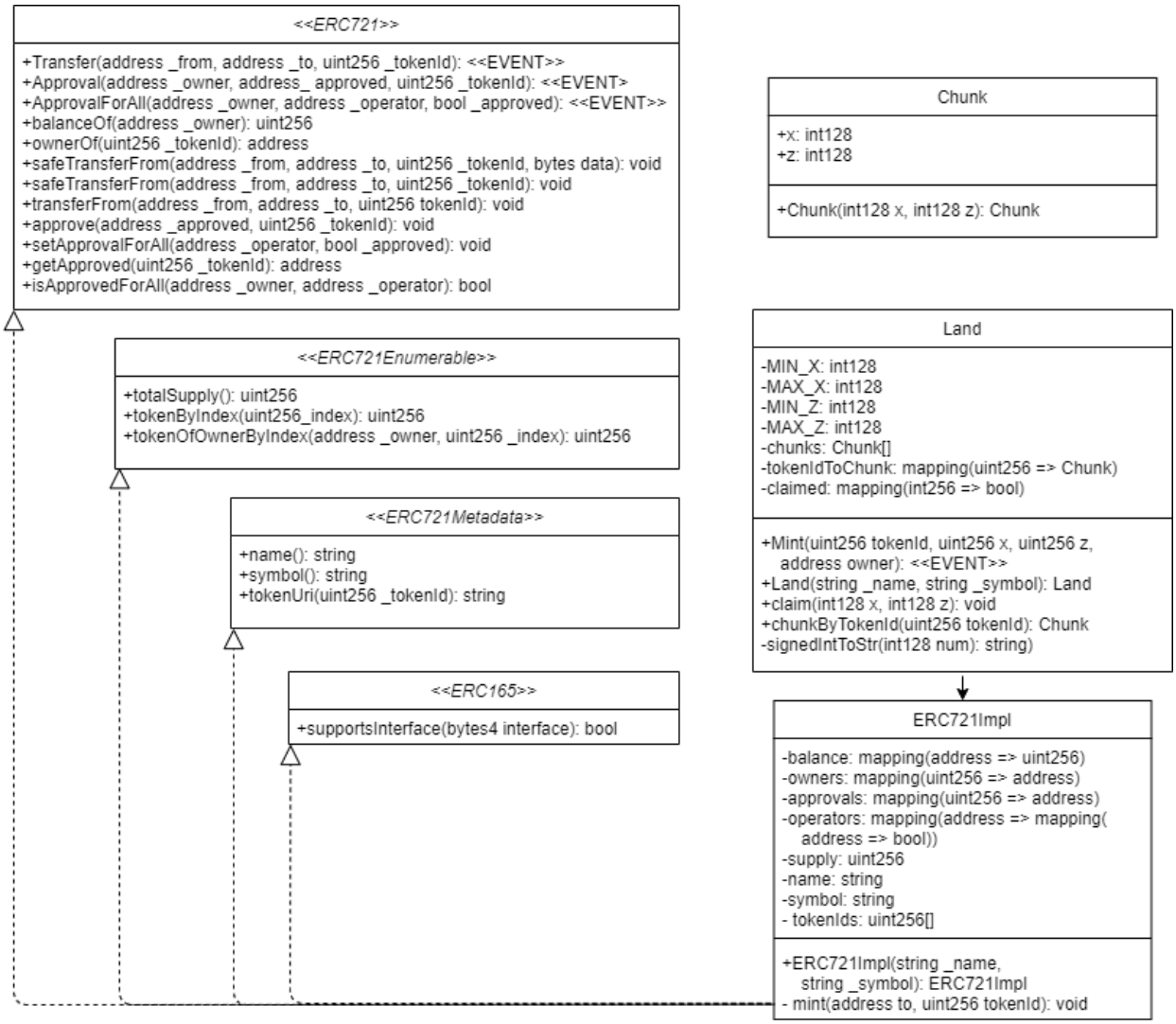
Why Blockchain?

The role of blockchain in this project is to provide decentralized and undisputed ownership of a digital asset (land in a Minecraft world) via non-fungible tokens. The NFTs act as deeds and give undisputed ownership of the corresponding chunk. The NFT cannot be taken from the holder without them initiating a transfer.

System Architecture

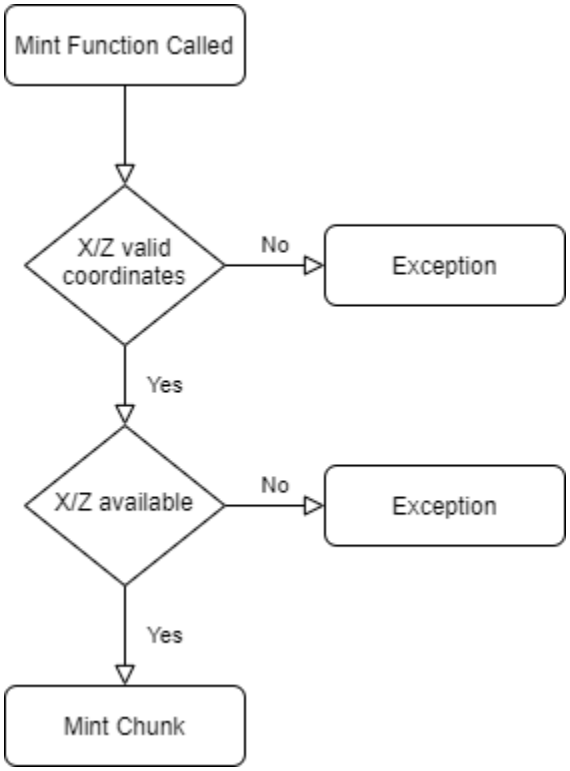


Smart Contract Design

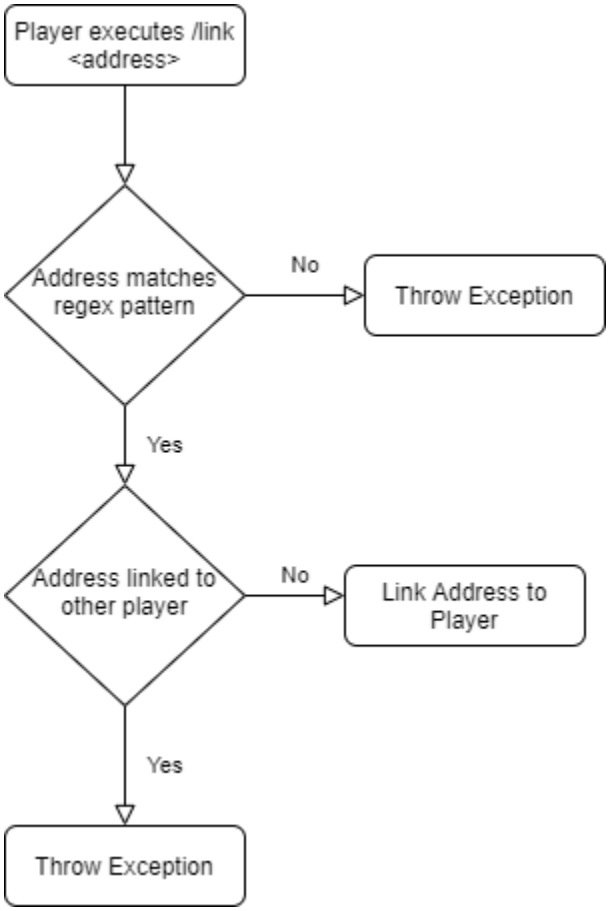


Workflows

Minting (Smart Contract)



Link Address (Server)



PlayerInteractEvent (Server)